

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A programmable processor comprising:
 - a data path capable of transmitting data;
 - an external interface operable to receive data from an external source and communicate the received data over the data path;
 - a register file containing a plurality of registers each having a register width, the register file coupled to the data path and ~~operable~~ configured to support processing of a plurality of threads and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;
 - an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute a plurality of instruction streams from the plurality of threads, each instruction stream including a single instruction that specifies an arithmetic operation to cause multiple instances of the arithmetic operation to be performed, each instance of the arithmetic operation to be performed using a different one of the plurality of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and
 - wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive,

in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation and execute the multiple instances of the arithmetic instruction to produce the catenated result.

2. (Original) The processor of claim 1 wherein the execution unit comprises a pipeline having a plurality of stages and wherein the pipeline interleaves execution of instructions from the plurality of instruction streams.

3. (Original) The processor of claim 2 wherein the pipeline is operable to simultaneously contain states of execution of at least two instructions from different instruction streams.

4. (Original) The processor of claim 2 wherein execution of the instructions is interleaved in a round-robin manner.

5. (Previously presented) The processor of claim 1 wherein the processor ensures only one thread from the plurality of threads can have an exception handled at any given time.

6. (Original) The processor of claim I further comprising a virtual memory addressing unit and a cache operable to store data communicated between the external interface and the data path.

7. (previously presented) The processor of claim 1 wherein the execution unit is further operable to, in response to decoding another single instruction specifying a first and a second register each containing a plurality of floating-point operands, multiply the plurality of floating-point operands in the first register by the plurality of floating-point operands in the second register to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a second catenated result.

8. (Currently amended) A programmable processor comprising:

a data path capable of transmitting data;

an external interface operable to receive data from an external source and communicate the received data over the data path;

first and second register files containing a plurality of registers each having a register width, the first and second register files coupled to the data path and ~~operable~~ configured to support processing of first and second threads, respectively, and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;

an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute first and second instruction streams from the first and second threads, respectively, the first and second instruction streams each including a single instruction that specifies an arithmetic operation to cause multiple instances of the arithmetic operation to be performed, each instance of the arithmetic operation to be performed using a different one of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and

wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the first register file and from the second register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation and execute the multiple instances of the arithmetic instruction to produce the catenated result.

9. (Original) The processor of claim 8 wherein the execution unit comprises a pipeline having a plurality of stages and wherein the pipeline interleaves execution of instructions from the first instruction stream with instructions from the second instruction stream.

10. (Original) The processor of claim 9 wherein the pipeline is operable to simultaneously contain states of execution of an instruction from the first instruction stream and an instruction from the second instruction stream.

11. (Original) The processor of claim 9 wherein execution of the instructions is interleaved in a round-robin manner.

12. (Previously presented) The processor of claim 9 wherein the execution unit is further operable to, in response to decoding another single instruction specifying a first and a second register each containing a plurality of floating-point operands, multiply the plurality of floating-point operands in the first register by the plurality of floating-point operands in the second register to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a second catenated result.

13. (Currently amended) A data processing system comprising:

(a) a bus coupling components in the data processing system;

(b) an external memory coupled to the bus;

(c) a programmable microprocessor coupled to the bus and capable of operation

independent of another host processor, the microprocessor comprising:

a data path capable of transmitting data;

an external interface operable to receive data from an external source and communicate the received data over the data path;

a register file containing a plurality of registers each having a register width, the register file coupled to the data path and ~~operable~~ configured to support processing of a plurality of threads and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;

an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute a plurality of instruction streams from the plurality of threads, each instruction stream including a single instruction that specifies an arithmetic operation to cause multiple instances of the arithmetic operation to be performed, each instance of the arithmetic operation to be performed using a different one of the plurality of data elements in partitioned fields of at least one of the registers to produce a catenated result; and

wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation and execute the multiple instances of the arithmetic instruction to produce the catenated result.

14. (Original) The system of claim 13 wherein the execution unit comprises a pipeline having a plurality of stages and wherein the pipeline interleaves execution of instructions from the plurality of instruction streams.

15. (Original) The system of claim 14 wherein the pipeline is operable to simultaneously contain states of execution of at least two instructions from different instruction streams.

16. (Original) The system of claim 14 wherein execution of the instructions is interleaved in a round-robin manner.

17. (Previously presented) The system of claim 13 wherein the processor ensures only one thread from the plurality of threads can have an exception handled at any given time.

18. (Priginal) The system of claim 13 further comprising a virtual memory addressing unit and a cache operable to store data communicated between the external interface and the data path.

19. (Previously presented) The system of claim 13 wherein the execution unit is further operable to, in response to decoding another single instruction specifying a first and a second register each containing a plurality of floating-point operands, multiply the plurality of floating-point operands in the first register by the plurality of floating-point operands in the second register to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a second catenated result.

20. (currently amended) A data processing system comprising:

(a) a bus coupling components in the data processing system;

(b) an external memory coupled to the bus;

(c) a programmable microprocessor coupled to the bus and capable of operation

independent of another host processor, the microprocessor comprising:

a data path capable of transmitting data

an external interface operable to receive data from an external source and communicate the received data over the data path;

first and second register files containing a plurality of registers each having a register width, the first and second register files coupled to the data path and ~~operable~~ configured to support processing of first and second threads, respectively, and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;

an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute first and second instruction streams from the first and second threads, respectively, the first and second instruction streams each including a single instruction that specifies an arithmetic operation to cause multiple instances of the arithmetic operation to be performed, each instance of the arithmetic operation to be performed using a different one of the plurality of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and

wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the arithmetic operation to be transmitted in parallel from the first register file and from the second register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the arithmetic operation and execute the multiple instances of the arithmetic instruction to produce the catenated result.

21. (Original) The system of claim 20 wherein the execution unit comprises a pipeline having a plurality of stages and wherein the pipeline interleaves execution of instructions from the first instruction stream with instructions from the second instruction stream.

22. (Original) The system of claim 21 wherein the pipeline is operable to simultaneously contain states of execution of an instruction from the first instruction stream and an instruction from the second instruction stream.

23. (Original) The system of claim 21 wherein execution of the instructions is interleaved in a round-robin manner.

24. (Previously presented) The system of claim 21 wherein the execution unit is further operable to, in response to decoding another single instruction specifying a first and a second register each containing a plurality of floating-point operands, multiply the plurality of floating-point operands in the first register by the plurality of floating-point operands in the second register to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a second catenated result.

25. (Previously presented) The processor of claim 1 wherein the arithmetic operation comprises an integer operation.

26. (Previously presented) The processor of claim 1 wherein the arithmetic operation comprises a floating-point operation.

27. (Currently amended) A programmable processor comprising:

a data path capable of transmitting data;

an external interface operable to received data from an external source and communicate the received data over the data path;

a register file containing a plurality of registers each having a register width, the register file coupled to the data path and operable configured to support processing of a plurality of

threads and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width.

an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute a plurality of instruction streams from the plurality of threads, each instruction stream including a single instruction that specifies a floating-point arithmetic operation to cause multiple instances of the floating-point arithmetic operation to be performed, each instance of the floating point arithmetic operation to be performed using a different one of the plurality of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and

wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the floating-point arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the floating-point arithmetic operation and execute the multiple instances of the floating-point arithmetic instruction to produce the catenated result.

28. (Currently amended) A data processing system comprising:

- (a) a bus coupling components in the data processing system;
- (b) an external memory coupled to the bus;
- (c) a programmable microprocessor coupled to the bus and capable of operation

independent of another host processor, the microprocessor comprising:

a data path capable of transmitting data;

an external interface operable to receive data from an external source and communicate the received data over the data path;

a register file containing a plurality of registers each having a register width, the register file coupled to the data path and ~~operable~~ configured to support processing of a plurality of threads and to store a plurality of multiple-bit data elements in partitioned fields, each of the multiple-bit data elements having an elemental width smaller than the register width;

an execution unit coupled to the data path, the execution unit ~~operable~~ configured to execute a plurality of instruction streams from the plurality of threads, each instruction stream including a single instruction that specifies a floating-point arithmetic operation to cause multiple instances of the floating-point arithmetic operation to be performed, each instance of the floating-point arithmetic operation to be performed using a different one of the plurality of multiple-bit data elements in partitioned fields of at least one of the registers to produce a catenated result; and

wherein each of the multiple-bit data elements has an elemental width, and the data path has a data path width multiple times greater than the elemental width, to allow multiple-bit data elements used for the multiple instances of the floating-point arithmetic operation to be transmitted in parallel from the register file to the execution unit, and wherein the execution unit is operable to receive, in parallel, multiple-bit data elements for the multiple instances of the floating-point arithmetic operation and execute the multiple instances of the floating-point arithmetic instruction to produce the catenated result.